UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

TP N°2: Expresiones regulares y expresiones regulares extendidas en C

(Trabajo práctico grupal)

Docente: Esp. Ing. Pablo D. Mendez

Comisión: K2055

Fecha de Entrega: 13/09/2025

Grupo: N°10

Link al repositorio grupal: https://github.com/utn-frba-ssl/25-055-10

Muñiz, Hernan Agustin	213.920-0	hmuniz@frba.utn.edu.ar	HernanMuniz1
Sanchez, Enzo Jose	213.232-1	ensanchez@frba.utn.edu.ar	EnzoJSanchez
Masia, Tomás	222.503-7	tmasiamoreno@frba.utn.edu.ar	TomasMasia
Schneider, Ivan	214.129-2	ivschneider@frba.utn.edu.ar	ivneider
Schvartzman, Lucas	214.130-9	lschvartzman@frba.utn.edu.ar	lucasschvartzman

Introducción

El objetivo principal de este programa es procesar el archivo de texto: *Breve_Historia.txt* aplicando distintas expresiones regulares utilizando la biblioteca estándar POSIX “regex.h” en C.

El propósito es analizar oraciones del texto y detectar patrones relevantes que permitan responder a las distintas consignas solicitadas.

Macros

Nuestro programa utiliza macros para parametrizar ciertos valores que pueden variar según el contexto de ejecución.

RUTA_ARCHIVO:
Define la ruta (path) del archivo de texto que será procesado.

En nuestro caso, el valor utilizado es "breve_historia.txt".

ORACION_MAX:
Establece la longitud máxima permitida para el almacenamiento de una oración durante el análisis.

En este trabajo se fijó en 1024 caracteres (valor conveniente para este texto).

En caso de querer cambiarlas, ir al archivo “main.h” en “/include”

Consideraciones sobre la implementación

Procesamiento secuencial del archivo:

El archivo no se carga completamente en memoria. Decidimos trabajar carácter por carácter, almacenando únicamente la oración en curso (en un buffer de tamaño ORACION_MAX). Si bien, el archivo tiene un tamaño relativamente pequeño (~63 KB), consideramos una buena práctica no asumir tal disponibilidad de memoria.

Delimitación de oraciones:

En un primer momento consideramos que cada punto (“.”) indicaría el fin de una oración. Sin embargo, detectamos ciertos casos para los cuales este criterio no es válido, por ejemplo:

1. Números: “4.000.000”
2. Abreviaturas de nombres propios: “Arturo U. Illia”.

Se observa que a pesar de utilizar el punto, no se trata de múltiples oraciones.

Para contemplar correctamente estos casos:

1. Implementamos una expresión regular para identificar abreviaturas de una sola letra mayúscula seguida de un punto (ej. “U.”, “A.”).

A través de un contador, podemos saber cuántas palabras abreviadas existen, que sería lo que debe restarse al total de puntos delimitadores de oración para conocer cuántas oraciones tiene el texto.

2. Se incorporó una condición en el bucle principal que valida que, para considerar un punto como delimitador de oración, el carácter siguiente sea un **espacio** o un **salto de línea**.

Con esto se descartan tanto los números, como el punto que aparece en el título seguido de un asterisco "".*

Durante la etapa de pruebas observamos diferencias en la forma en que se procesaban los saltos de línea entre entornos de ejecución basados en UNIX (WSL) y no basados en UNIX (OnlineGDB).

Tras investigar el origen del problema, identificamos que se debía a la representación de los finales de línea: en sistemas UNIX se utiliza "\n", mientras que en otros entornos aparece "\r".

Para garantizar la correcta detección de los delimitadores de oración en ambos casos, nuestra solución fue extender la condición del bucle principal, de manera tal que el carácter siguiente al punto pueda ser un espacio (' '), un salto de línea ('\n'), un retorno de carro ('\r') o el fin de archivo (EOF).

Con esta adaptación, confirmamos que los resultados obtenidos son consistentes en ambas plataformas de prueba, eliminando las discrepancias detectadas inicialmente.

Verificación de resultados

Para garantizar que nuestro algoritmo cuenta correctamente las oraciones, se realizó un conteo manual sobre el archivo.

El resultado fue de 409 oraciones, valor que coincide con el obtenido por el programa.

Limitaciones conocidas

Las siguientes limitaciones se deben a que, en el texto indicado, no era necesario contemplar dichos casos:

- El tamaño máximo de cada oración está delimitado por la macro ORACION_MAX, aunque esta puede ser modificada en caso de ser necesario (ver Macros).
- No se contemplan casos de abreviaturas de más de una letra (como “Dr.” o “Ing.”).
- No se contemplan casos como el de puntos suspensivos.
- No se contemplan finales de oración con signos de exclamación o interrogación.